REMARKS/ARGUMENTS

In response to the Office Action dated June 29, 2009, claims 1 to 15, 17 to 21, and 23 to 27 are presented for examination, of which claims 1, 6, 10, 20, 23 and 26 are independent. Reconsideration and further examination are respectfully requested.

Claim Rejections – 35 USC § 102

Claims 1-15, 17-21, and 23-27 are rejected under 35 USC § 102(e), as being anticipated by Dye, et al. (US 6,518,965 B2). The rejections are respectfully traversed.

The background discussion of the present application notes that a problem arises when a specific command needs to be executed at a specific time or in relation to another executed command. Often times, when a specific command or external event occurs, the system will seek to execute a related command usually related to the previous command or the external event. In a conventional command processing system, the first-in first-out processing steps prevent the related command from being processed until all other commands within the queue or buffer are executed. Therefore, there exists a need for a command processing system whereupon the occurrence of multiple events may be properly synchronized without reducing system efficiency. The claimed apparatus achieves this goal by providing a mechanism for commands in a real time event command buffer to be fetched and executed in response to a real time event.

Independent claim 1 recites:

A method for processing command information in a command processing system, the method comprising:

(a) detecting a real time event while monitoring a plurality of event signals, wherein the plurality of event signals are generated by a plurality of engines and one of the plurality of engines is a 3D engine; and

(b) causing commands in a real time event command buffer to be fetched and consumed in response to the real time event.

Dye is not seen to disclose or suggest "detecting a real time event while monitoring a plurality of event signals" and "causing commands in a real time event command buffer to be fetched and consumed in response to the real time event."

Rather, the disclosure in Dye is for an apparatus that renders 3-D geometry (contained in a 3-D VDRL) into pixel data and subsequently passes the pixel data to a conventional frame buffer or rendering engine for real time display. Nowhere does Dye discuss the detection of real time events or the fetching and processing of commands in response to a real time event as claimed in the current application.

In support of the rejection, examiner has cited the following disclosure from Dye:

> The IMC includes a novel system and method for manipulating and rendering 3D graphics. The IMC first operates to construct a 3-D Virtual display refresh list (3-D VDRL) in system memory. The IMC then executes the 3-D VDRL by reading the VDRL and generating and/or accessing the appropriate pixel data from system memory to construct an image or display objects. For example, execution of the 3-D VDRL causes the generated pixel data to be stored in memory as an image. Dye, column 4, lines 30-38.

None of the above cited operations performed by the IMC constitutes detection of real time events and the fetching and processing of commands in response to a real time event as claimed in the current application. Rather, construction of the 3-D VDRL by the IMC in system memory involves creation of a specialized display list data structure containing 3-D geometry (the 3-D VDRL). The IMC subsequently executes the 3-D VDRL to generate pixel data which is stored in memory as an image. But, the execution of the 3-D VDRL by the IMC does not occur in response to a real time event. Rather, Dye states that "the IMC then executes the 3-D VDRL" subsequent to constructing it. This does not disclose or suggest that the 3-D VDRL is executed in response to a real time event.

The examiner also cites the following in support of the rejection:

> The IMC stores the pixel data generated by execution of the 3-D VDRL in a conventional frame buffer and performs a refresh of the frame buffer to render the pixel data on the screen. The IMC may also provide the pixel data directly to a rendering engine for real time display after executing the 3-D VDRL. Dye, column 4, lines 44-49.

As discussed above and in Dye, the 3-D VDRL is a display list containing 3-D geometry. Execution of this display list results in the generation of pixel data which is stored in a

conventional frame buffer. The above cited section from Dye discloses that the conventional frame buffer can render the pixel data to the screen. This operation involves reading the pixel data line by line from the frame buffer and drawing it on the screen. However, this operation does not involve fetching and processing of commands in response to a real time event as claimed in the current application. The above cited section of Dye further discloses that the pixel data may be transferred directly to a rendering engine for real time display. There is no meaningful difference between the rendering of pixel data to the screen by a conventional frame buffer and the real time display of pixel data by the rendering engine. That is, the real time display of pixel data by the rendering engine also involves reading the pixel data line by line from the frame buffer and drawing it on the screen. As with the drawing of pixel data from the frame buffer, this operation does not involve fetching and processing of commands in response to a real time event as claimed in the current application.

The examiner also cites the following in support of the rejection:

> The 3-D VDRL is read and interpreted for triangle parameter data, texture address, attributes and other control information. The 3D graphics engine then renders all the triangle segments for multiple triangles for an entire span line as indicated in the 3-D VDRL. During execution of the 3-D VDRL, the 3-D VDRL is fetched for instructions and pointers to control the rasterization of the output image by the 3D engine. Execution of the 3-D VDRL may cause rasterization of the output image to the memory on to the display device. Dye, column 5, lines 19-29.

This disclosure from Dye discusses how the 3-D VDRL is processed to generate an output image which is subsequently made available to a display device. The statement that "execution of the 3-D VDRL may cause rasterization of the output image to the memory on to the display device" means that subsequent to traversal of the display list and generation of the pixel data, the pixel data is transferred to the display device. This operation does not involve fetching and processing of commands in response to a real time event as claimed in the current application.

For the reasons discussed above, independent claim 1 is believed to be allowable over the cited reference. Independent claims 6, 10, 13, 20, 23, and 26 generally correspond to the method of claim 1 and are believed to be allowable for at least the same reasons. The remaining claims in the application are each dependent from one of the aforementioned independent claims and are also believed to be allowable for at least the same reasons.

CONCLUSION

In light of the amendments contained herein, Applicants submit that the application is in condition for allowance, for which early action is requested.

Please charge any fees or overpayments that may be due with this response to Deposit Account No. 17-0026.

Respectfully submitted,

Dated: August 26, 2009

By: Matthew J. Evans, Reg. No. 56,530

QUALCOMM Incorporated
Attn: Patent Department
5775 Morehouse Drive
San Diego, California 92121-1714
Telephone: (858) 651-7571
Facsimile: (858) 845-3983